# YAWL Engine User Manual

***Beta – 7 Release***

# Document Control

| Date | Author | Version | Change |
|---|---|---|---|
| 25 Aug 2004 | Sean Kneipp | 0.1 | Initial Draft |
| 30 Aug 2004 | Sean Kneipp | 1.0 | Final Draft |
| 4 Feb 2005 | Guy Redding | 1.1 | Added YAWL XForms details |
| 4 Feb 2005 | Lachlan Aldred | 1.2 | Updated Issues & 3rd-Party Software Acknowledgements |
| 12/04/2005 | Lachlan Aldred | 1.3 | Updated troubleshooting section. |
| 22/12/2005 | Michael Adams | 1.7 | Updated for Beta 7 Version. Document version brought in line with Release Version |

# Contents


Document Control ..... ii
Welcome to YAWL ..... 1
What is YAWL? ..... 1
Obtaining the Latest Version of the YAWL Engine ..... 1
The YAWL Project ..... 2
YAWL Architecture ..... 2
Getting Started ..... 3
Overview ..... 3
Launching the YAWL Engine ..... 3
The YAWL Workspace ..... 4
Key YAWL Pages ..... 4
YAWL Administration ..... 5
Loading a YAWL specification ..... 5
Launching a YAWL Specification ..... 6
Unloading a YAWL Specification ..... 7
Managing User Access ..... 7
Data Elements of a Case ..... 8
Data Perspective ..... 8
Providing Data for the Case Using a Form ..... 8
Providing Data for the Case Using XML ..... 9
How to Input the Data Using XML ..... 10
Work Items ..... 11
Viewing Available Work ..... 11
Checking Out Available Work ..... 11
Saving Detailed Information for a Work Item ..... 12
Submitting a Work Item ..... 14
Creating a New Instance of a Work Item ..... 14
Suspending a Work Item ..... 15
Known Issues ..... 16
Engine (`yawl.war`) ..... 16
Worklist (`worklist.war`) ..... **Error! Bookmark not defined.**
Web Service Invoker (`wsInvoker.war`) ..... 16
YAWL XForms Processor (`YAWLXForms.war`) ..... 16
Troubleshooting ..... 17
Third–Party Software Acknowledgements ..... 18
Acknowledgements ..... 22

# Welcome to YAWL

## *What is YAWL?*

Based on a rigorous analysis of existing workflow management systems and workflow languages, we have developed a new workflow language: YAWL (Yet Another Workflow Language). To identify the differences between the various languages, we have collected a fairly complete set of workflow patterns. Based on these patterns we have evaluated several workflow products and detected considerable differences in expressive power. Languages based on Petri nets perform better when it comes to state-based workflow patterns. However, some patterns (e.g. involving multiple instances, complex synchronizations or non-local withdrawals) are not easy to map onto (high-level) Petri nets. This inspired us to develop a new language by taking Petri nets as a starting point and adding mechanisms to allow for a more direct and intuitive support of the workflow patterns identified.

YAWL can be considered a very powerful workflow language, built upon experiences with languages supported by contemporary workflow management systems. While not a commercial language itself it encompasses these languages, and, in addition, has a formal semantics. Such an approach is in contrast with e.g. WfMC's XPDL which takes commonalities between various languages as a starting point and does not have formal semantics. Its design hopefully allows YAWL to be used for the purposes of the study of expressiveness and interoperability issues.

At this stage YAWL supports the control-flow perspective and the data perspective. However, it is important that relations with the resource perspective are investigated and formalized. For YAWL to be more applicable in the area of web services and Enterprise Application Integration, it is also desirable that support for communication patterns is built-in.

## *Obtaining the Latest Version of the YAWL Engine*

As new versions of the YAWL Engine are released to the public, they will be available for download at the YAWL website:

http://www.yawl-system.com

Developers interested in obtaining the source code for the YAWL Engine, can download the files from here:

http://sourceforge.net/projects/yawl

## *The YAWL Project*

For more information and progress on the YAWL project, visit the YAWL Homepage on Queensland University of Technology's Centre for Information Technology Innovation website:

http://www.yawl-system.com

## *YAWL Architecture*

The following image, **Fig.1.**, depicts the interaction between components of the YAWL Engine.

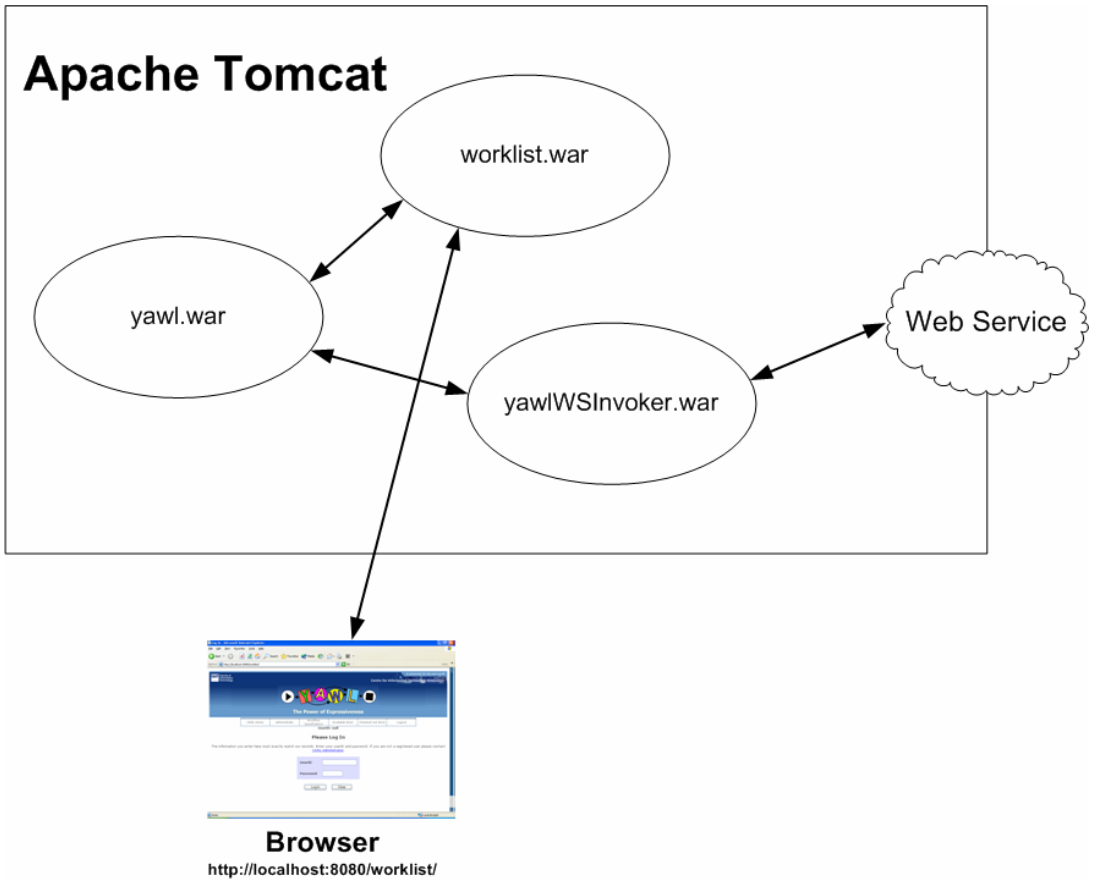


**Fig.1. The YAWL Engine Components**

# Getting Started

## *Overview*

Before you jump straight into the YAWL Engine and start testing your favourite workflows, you will need to make sure that the Tomcat web-service has been started. YAWL is a browser-based web application that requires a server jsp environment, such as Apache Tomcat.

For full instructions on starting the Tomcat service, or obtaining the Apache Tomcat software, please consult the YAWL Installation Manual.

## *Launching the YAWL Engine*

1. Open up your web browser.
2. Type the following address into the address line and press enter (see **Fig.2.**):

   http://localhost:8080/worklist.

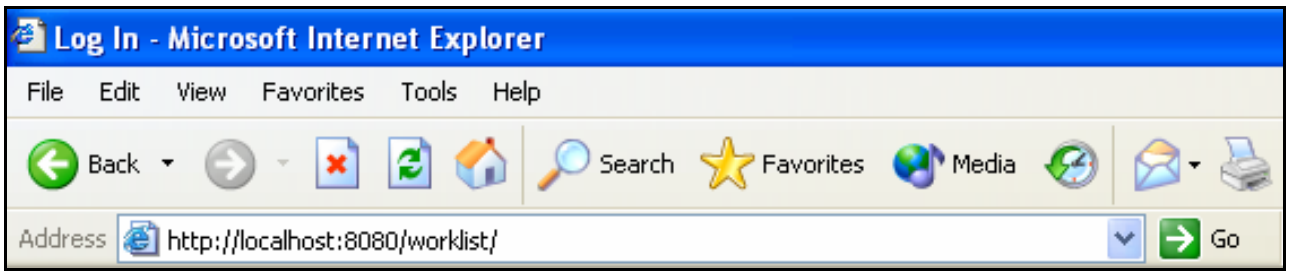


**Fig.2. Navigate to the YAWL Log In webpage**

3. The YAWL Log In screen will be displayed
4. Log in with the following details (case sensitive):

   UserID: `admin`

   Password: `YAWL`

## *The YAWL Workspace*

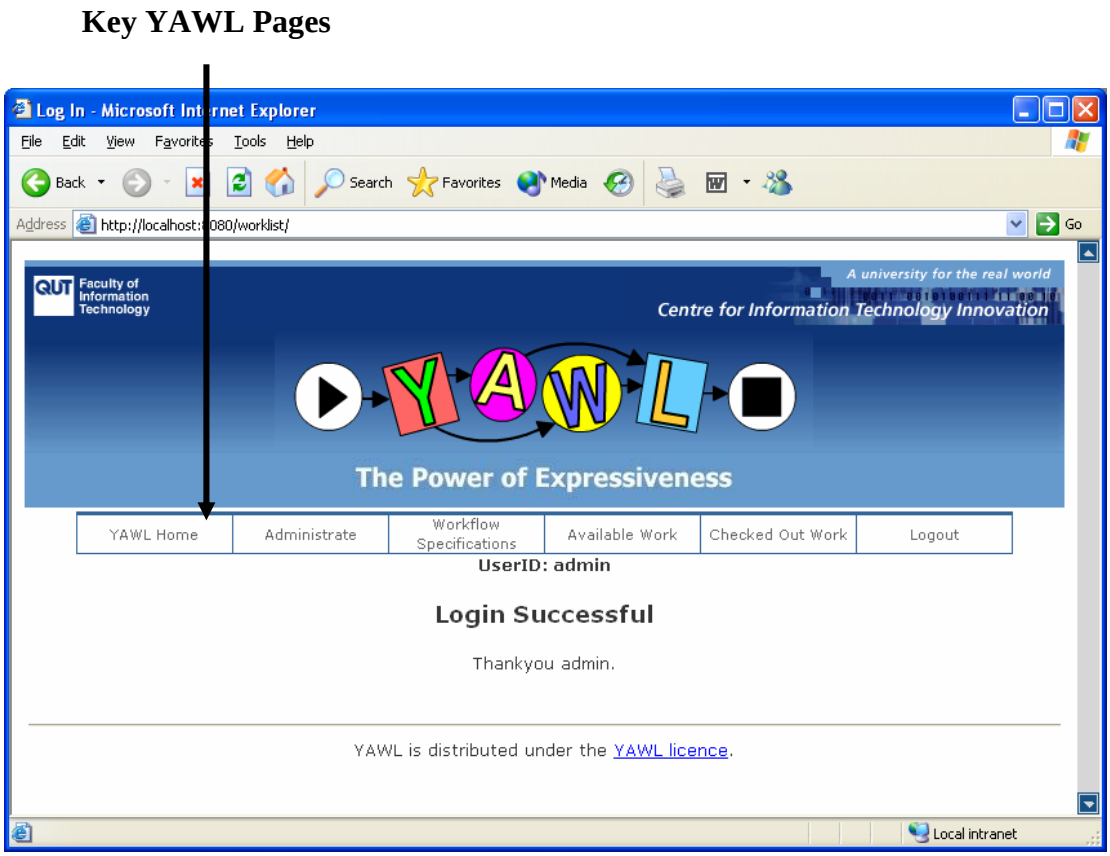


**Fig.3. YAWL Workspace**

### Key YAWL Pages

The separate pages of the browser-based application, representing the different functions of the YAWL Engine:

- YAWL Home
- Administrate
- Workflow Specifications
- Available Work
- Checked Out Work
- Logout

# YAWL Administration

## *Loading a YAWL specification*

In additional to a graphical syntax, the YAWL language has an XML syntax. The runtime environment requires this YAWL XML syntax. To load a YAWL specification:

1. Navigate to the Administrate page:

   http://localhost:8080/worklist/admin.

2. In the Manage Specifications section of the page, Browse for a YAWL specification (.xml file) and choose Open.

   *The installation package for YAWL contains a number of sample YAWL specifications (XML files). The specification used throughout this document refers to the "makeTrip3.XML" file.*

3. Click on Upload.

4. You should now see the details of your specification. **Fig.4.**

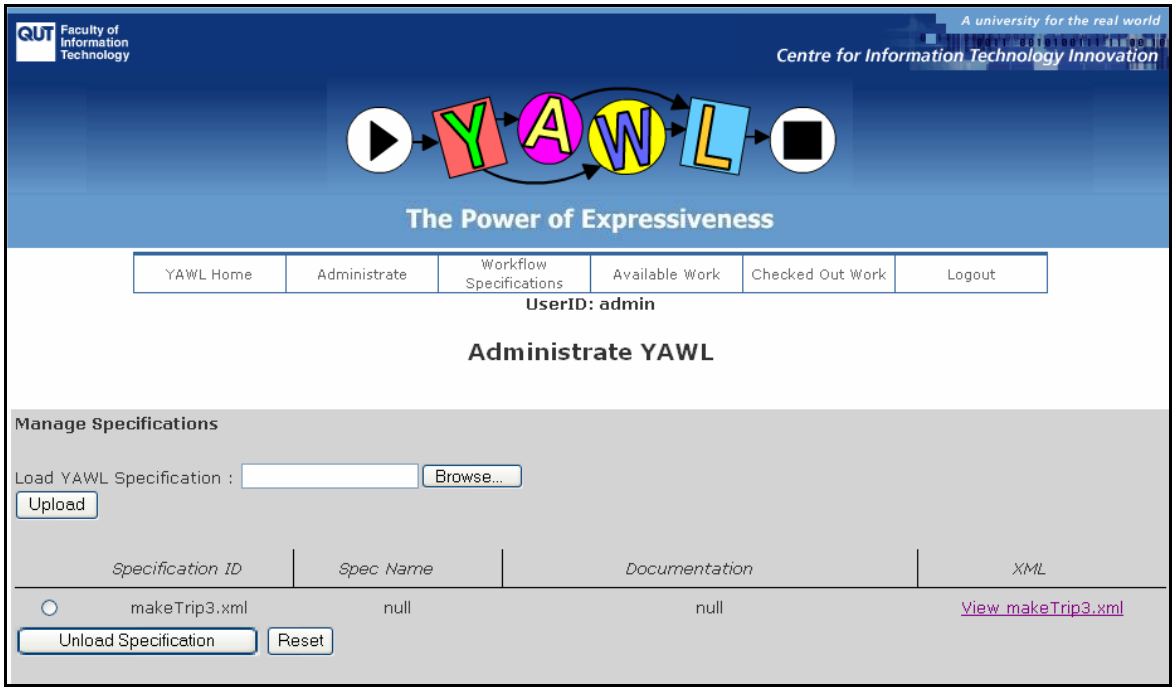


**Fig.4. Upload YAWL Specification**

## *Launching a YAWL Specification*

In order to kick-off the workflow specification, you need to 'launch a case'. To launch a YAWL specification that has already been uploaded into the YAWL Engine:

1. Navigate to the Workflow Specifications page:

   http://localhost:8080/worklist/viewSpecifications.

2. In the Active YAWL Specifications section of the page, choose the specification to launch.

3. Click on the Launch Case button. **Fig.5.**

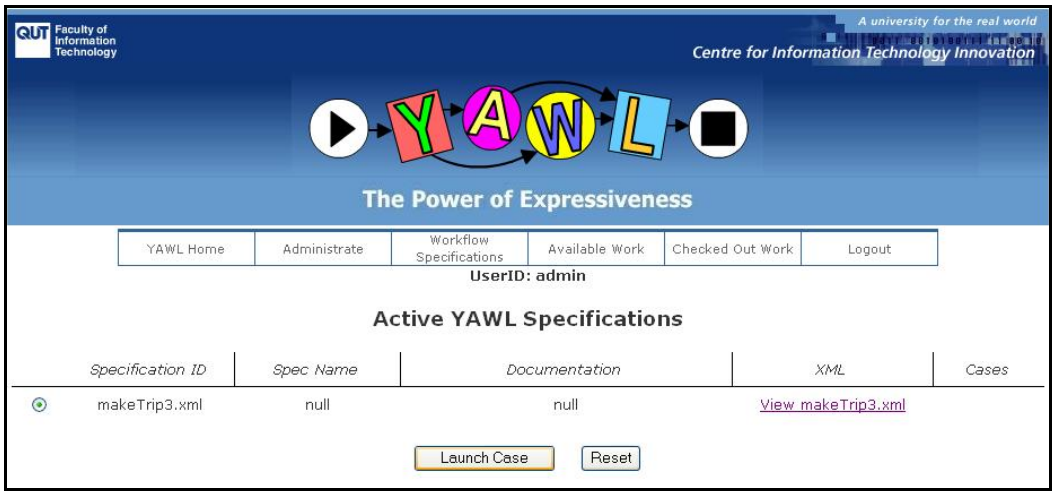


**Fig.5. Launching the Case**

4. Then in the following Launch Case section of the page, click on the Start Case button. **Fig.6.**

   Depending on the specification loaded, you may be required to launch the case with specific information for data elements. For more information on providing data throughout the case, consult the **Data Elements of a Case** section of this manual.

**Fig.6. Starting the Case**

5. You will then see a confirmation screen to indicate that a case has been launched with a specific identification number.

## *Unloading a YAWL Specification*

You can remove a YAWL specification so that it is no longer available to be started. To remove a specification:

1. Navigate to the Administrate page:

   http://localhost:8080/worklist/admin.

2. In the Manage Specifications area of the Content Pane, choose the specification to remove.

3. Click on the Unload Specification button.

## *Managing User Access*

Alternate user accesses to the YAWL Engine can be created through the Administrator. To create a new user access:

1. Navigate to the Administrate page:

   http://localhost:8080/worklist/admin.

2. In the Manage Users section of the page, enter the new user's Username and Password, then retype the Password in the Confirm Password box. **Fig.7.**

3. To create the user under the Administrator role, click the Create Button, otherwise click the Create User button.

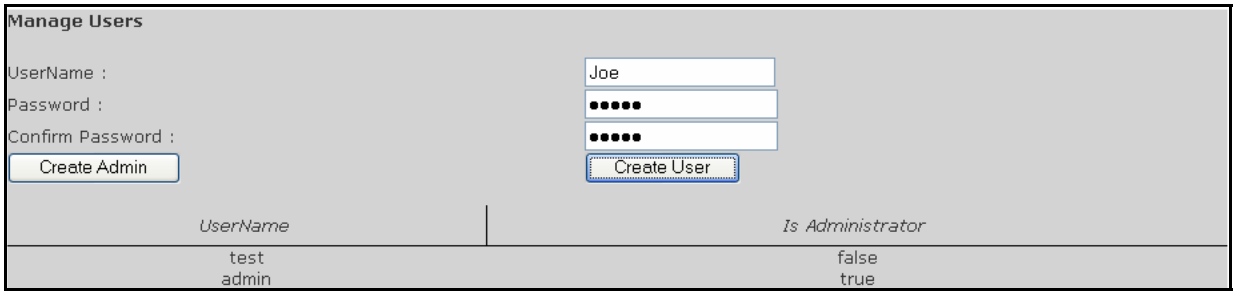


**Fig.7. Managing Users**

# Data Elements of a Case

## *Data Perspective*

Although the initial focus of YAWL was on control flow, it has been extended to offer full support for the data perspective. It is possible to define data elements and use them for conditional routing, for the creation of multiple instances, for exchanging information with the environment, etc. Most of the existing workflow management systems use a propriety language for dealing with data. YAWL is one of the few languages that completely relies on XML-based standards like XPath and XQuery.

## *Providing Data for the Case Using a Form*

During certain activities of a YAWL Case, you may be required to provide some information for data elements that have been established in the XML specification. Requests for data can occur when "Launching a Case" or when editing "Checked Out Work" items.

**Fig.8.** depicts a request for data form for multiple data elements, while performing an edit of a "Checked Out Work" item. The red star next to a field indicates that input is required for that field. The form submits data back to YAWL only when all data in the forms is valid.

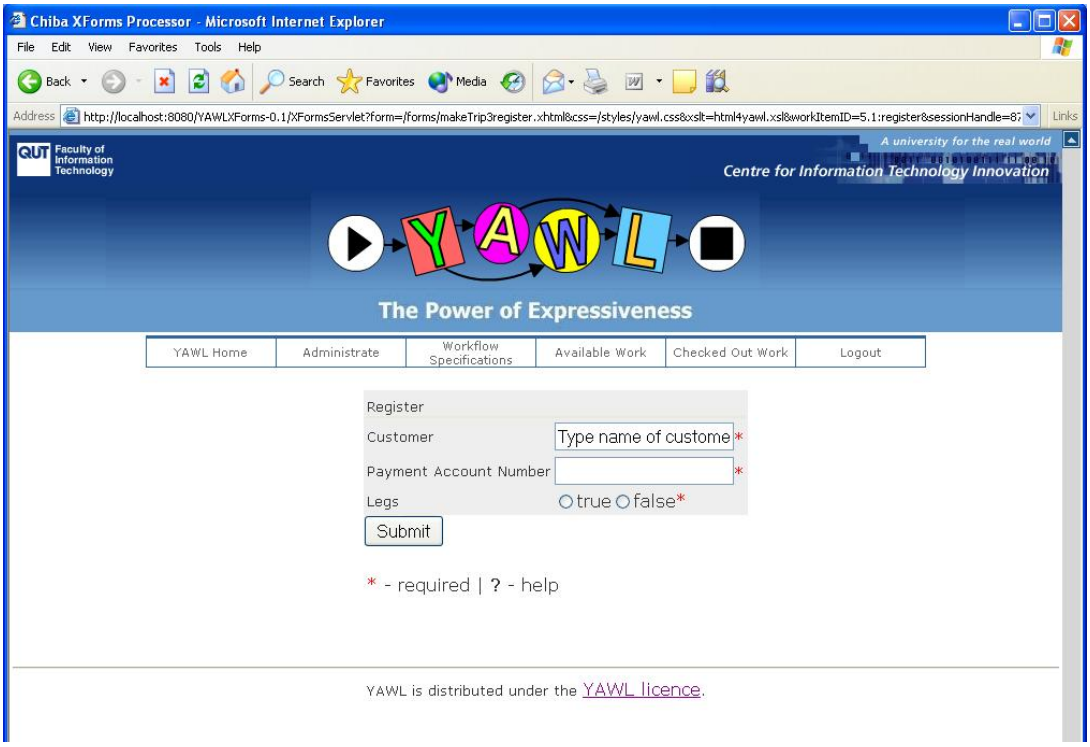


**Fig.8. Request for Data using a Form**

## *Providing Data for the Case Using XML*

It is also always possible to enter data using XML. This method of user input is useful if data input using a form has failed for any reason. **Fig. 9** is the XML input page. To get to this page *click the link* for the work item ID on the Checked Out Work Items page (**Fig. 12**) instead of selecting the radio button for a Work Item and clicking the Edit Work Item button.

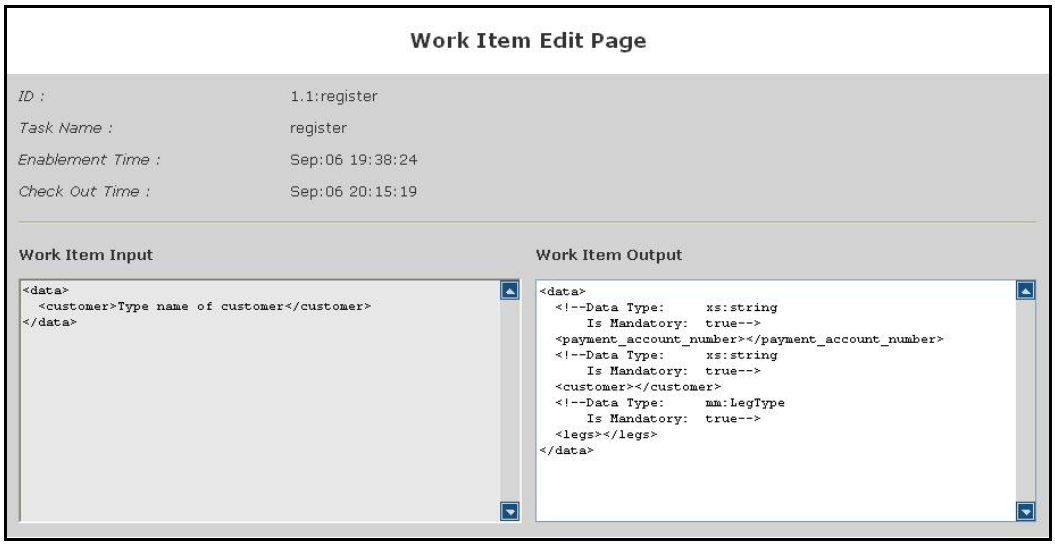


**Fig. 9 Request for Data using XML**

## How to Input the Data Using XML

The structure of the request for data follows the XML format of opening and closing "tags". The data is entered between the open and close tags that refer to that data element.

In the "makeTrip3.xml" specification, the example below (**Fig. 10.**) depicts a request to enter the name of the customer going on a trip, for the `customer` data element.

1. Firstly the **`data`** tag is opened.
2. Then the **`Data Type`** tag (or comments tag) is opened to let the user know the details of the data required.
3. The `Data Type` tag is requesting a **`string`** variable from the user and the information required is Mandatory. The tag is closed.
4. Then the **`customer`** tag is opened, referring to the data element to store the information.
5. The user then enters the **customer name** within the customer tags.
6. Finally the **`data`** tag is closed.

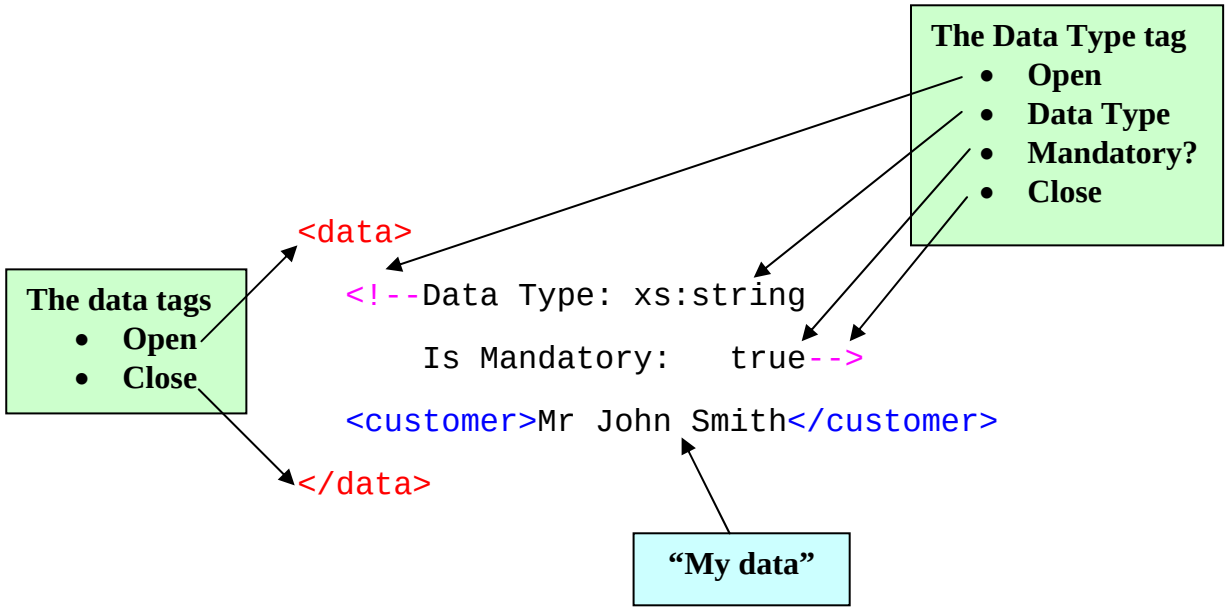


**Fig. 10. Inputting the Data**

# Work Items

## *Viewing Available Work*

Alternate user accesses to the YAWL Engine can be created through the Administrator. To create a new user access:

1. Navigate to the Available Work page:

   http://localhost:8080/worklist/availableWork.

2. All available work items will be listed, according to the specification loaded. Work items can be related back to a specific instance of a case, through their case number. **Fig.11.**

**Fig. 11. Available Work Items**

## *Checking Out Available Work*

When a person is ready to start working on an available work item, the item is "Checked Out".

1. Navigate to the Available Work page:

   http://localhost:8080/worklist/availableWork.

2. Click on the radar button next to the work item and click the button "Check Out"

## *Saving Detailed Information for a Work Item*

While a person is working on an item that has been "Checked Out", specific details may be required to be saved against that item.

Information saved against a work item can be used to launch other YAWL cases or provide data for other interfacing systems. For more information on providing data in work items, consult the **Data Elements of a Case** section of this manual.

1. Navigate to the Checked Out Work page:

   http://localhost:8080/worklist/checkedOut.

2. Click on the radar button next to the work item and click the button "Edit Work Item" **Fig.12.**

**Fig. 12. Checked Out Work Items**

3. Enter any required data using a form for the data elements that have been requested, in the Work Item Output section of the page and click Save Work Item. **Fig.13.**

4. Alternatively, click on the link for the Work Item ID to enter any required data for the requested data elements using XML. **Fig. 14.**

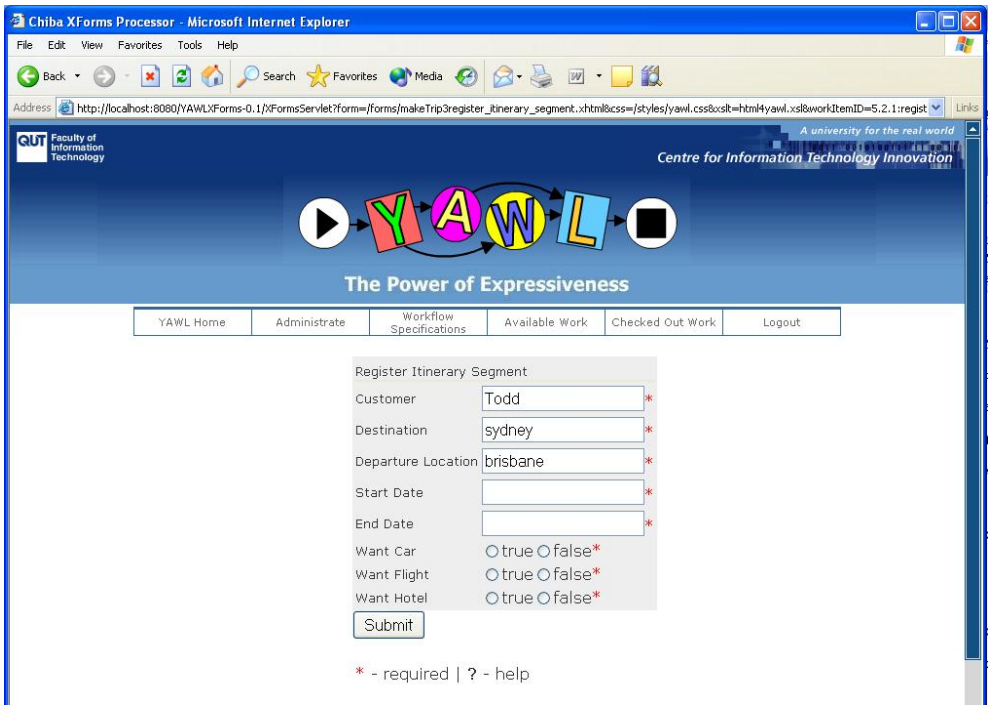


**Fig 13. Work Item Edit Page Using a Form**

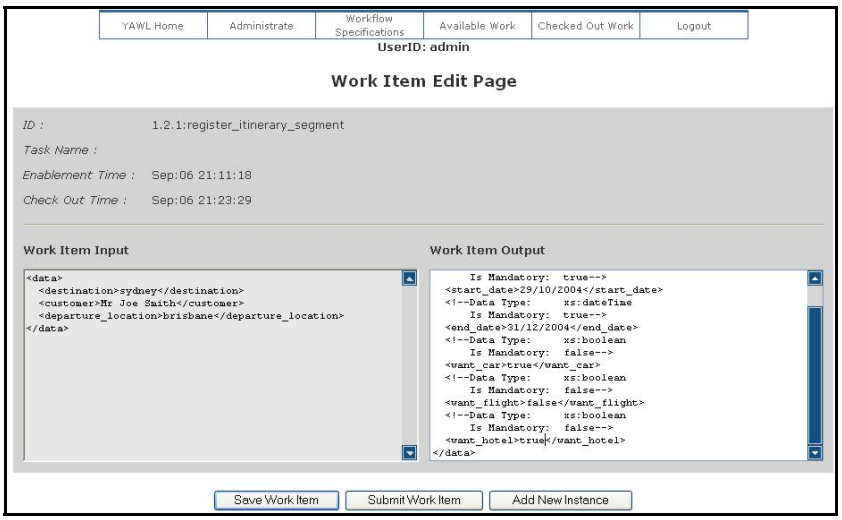


**Fig. 14. Work Item Edit Page Using XML**

## *Submitting a Work Item*

Once a work item has been completed, it must be submitted for the token to be advanced in the workflow.

1. Navigate to the Checked Out Work page:

   http://localhost:8080/worklist/checkedOut.

2. Click on the radar button next to the work item and click the button "Edit Work Item" **Fig.12.**

3. Enter any required data for the data elements that have been requested in the Work Item Output section of the page. **Fig.13, 14.**

4. Click on the Submit Work Item button.

## *Creating a New Instance of a Work Item*

If the specification supports it, new instances of a work item can be created as an item is "checked out", reflecting multiple instance tasks in the workflow.

*The current example specification, "makeTrip3.xml" does not contain any multiple instance tasks, so instructions below refer to the specification "MakeMusic.XML".*

1. Navigate to the Available Work page:

   http://localhost:8080/worklist/availableWork.

2. Check Out a work item that supports multiple instance creation.

   *The "MakeMusic.XML" specification contains a multiple instance task on the work item "Write a Song".*

5. Navigate to the Checked Out Work page:

   http://localhost:8080/worklist/checkedOut.

3. Click on the radar button next to the work item and click the button "Add New Instance" **Fig.12.**

4. Enter any required data for the data elements that have been requested and click on Create Instance. **Fig.15.**

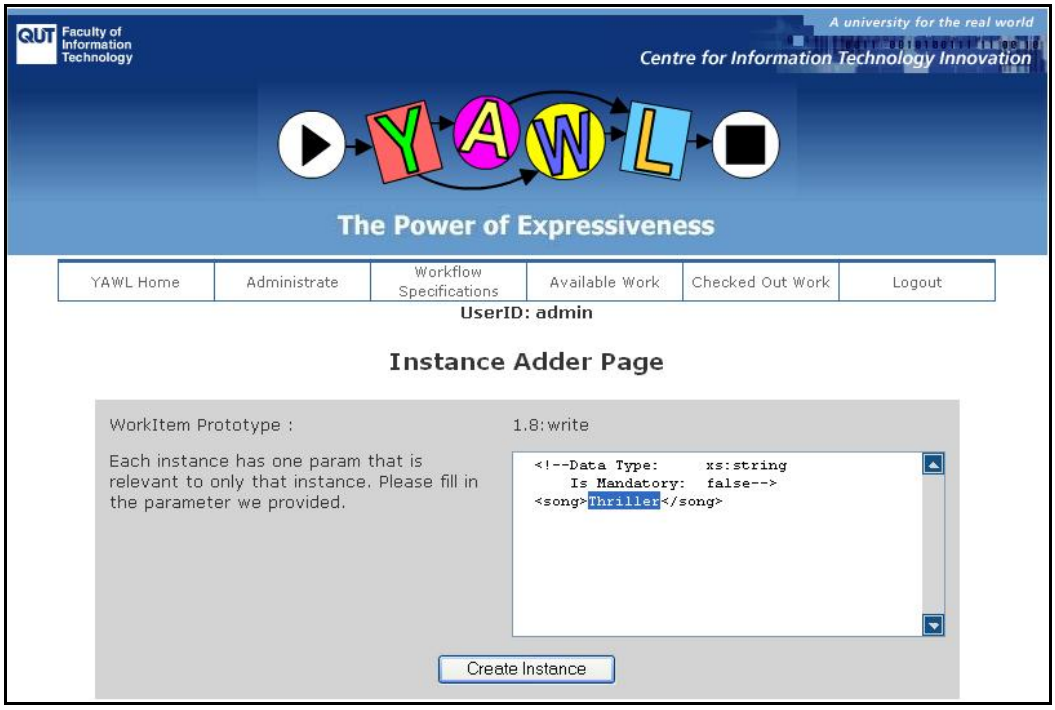


**Fig.15. Creating a New Instance of a Work Item**

Your new instance of the Work Item will be now be listed as Available Work.

## *Suspending a Work Item*

Work items that have been "checked out" to a person, can be returned to the available work items. All data that has been saved for that work item remains intact.

5. Navigate to the Checked Out Work page:

   http://localhost:8080/worklist/checkedOut.

6. Click on the radar button next to the work item and click the button "Suspend Task" **Fig.12.**

# Known Issues

## *Engine (`yawl.war`)*

## *Web Service Invoker (`wsInvoker.war`)*

- The web service invoker does not support sending call back addresses to an asynchronous web service.
- The web service invoker component is not able to invoke web services with complex data types as their parameters.

## *YAWL XForms Processor (`YAWLXForms.war`)*

- All fields seen on the forms are editable. This means that input parameters on a form are editable. Input parameters should not be editable.
- Currently forms cannot be generated for schemas that contain enumerated types.

# Troubleshooting

### Error in Uploading a Specification

When uploading a specification, the YAWL Engine performs three different sets of validation for the specification. Please check that your specification passes the following three validations:

- Schema Validation
- XML Validation
- YAWL Engine Application Validation.
  *This validation is particular to the YAWL application and can represent inconsistencies in the workflow, for example, if a task is defined in the specification but doesn't appear between the input and output condition.*

### Specification with an identical id.

This error occurs when you are uploading a specification that has already been uploaded into the engine. If you have made changes to the specification and want to upload a newer version, then Unload the existing specification before uploading the newer version.

# Third–Party Software Acknowledgements

(a) Software developed by the JDOM Project (http://www.jdom.org/),

JDOM Beta 9, Copyright © 2000-2003 Jason Hunter & Brett McLaughlin. All rights reserved. This software is subject to the terms of the licence available at http://www.jdom.org/.

(b) Software developed by the Apache Software Foundation (http://www.apache.org/) (Xerces2 Java Parser 2.6.0 Release) (http://xml.apache.org/xerces2-j/index.html), Copyright © 1999-2003 The Apache Software Foundation. All rights reserved. This software is subject to the terms of the licence available at http://xml.apache.org/.

Web Services Invocation Framework 2.0.- Release (http://ws.apache.org/wsif/), Copyright © 1999-2003 The Apache Software Foundation. All rights reserved. This software is subject to the terms of the licence available at http://xml.apache.org.

(c) The Saxon version 7 XLST Processor from Michael Kay (http://saxon.sourceforge.net/). This software is subject to the Mozilla Public License Version 1.0 and may not be used except in compliance with that License (a copy is available at http://www.mozilla.org/MPL/). The source code version of this software is available under the terms of that License.

(d) Chiba XForms processor version 0.9.9, available at http://sourceforge.net/projects/chiba/

(e) Hypersonic SQL

```
Hypersonic SQL is distributed under the following licence:

COPYRIGHTS AND LICENSES

ORIGINAL LICENSE (a.k.a. "hypersonic_lic.txt")

For content, code, and products originally developed by Thomas
Mueller and the Hypersonic SQL Group:

Copyright (c) 1995-2000 by the Hypersonic SQL Group.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following
conditions are met:

Redistributions of source code must retain the above copyright
notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above
copyright notice,
this list of conditions and the following disclaimer in the
```

```
documentation
and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its
contributors may be used to endorse or promote products
derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP,
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the
Hypersonic SQL Group.

For work added by the HSQL Development Group (a.k.a.
hsqldb_lic.txt):

Copyright (c) 2001-2004, The HSQL Development Group
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following
conditions are met:

Redistributions of source code must retain the above copyright
notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above
copyright notice,
this list of conditions and the following disclaimer in the
documentation
and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names
of its
contributors may be used to endorse or promote products
derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP,
HSQLDB.ORG,
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

(f) XSD and EMF (source available at http://www.eclipse.org/xsd)

XSD and EMF are made available under the Common Public License:

Common Public License Version 1.0
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.
1. DEFINITIONS
"Contribution" means:
a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
i) changes to the Program, and
ii) additions to the Program;
where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.
"Contributor" means any person or entity that distributes the Program.
"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.
"Program" means the Contributions distributed in accordance with this Agreement.
"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.
2. GRANT OF RIGHTS
a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.
c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.
3. REQUIREMENTS
A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:
a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:
a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.
Contributors may not remove or alter any copyright notices contained within the Program.
Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.
4. COMMERCIAL DISTRIBUTION
Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.
For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.
5. NO WARRANTY
EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.
6. DISCLAIMER OF LIABILITY
EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
7. GENERAL
If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.
All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.
Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new

versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.
This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

# Acknowledgements

This documentation includes feedback from Lachlan Aldred and Marlon Dumas, Queensland University of Technology.